

JAPANESE PATENT APPLICATION LAID-OPEN PUBLICATION

NO.: 3-265027

PUBLISHED DATE: November 26, 1991

PATENT APPLICATION NO.: 2-62650

5 FILED DATE: March 15, 1990

Page 201, lower left column, line 18 to next column, line 17

The foregoing embodiment has been described with reference
10 to an example in which the image file 10 is provided in the
auxiliary storage 3. However, as shown in FIG. 3, the main
storage 4 may be divided into two parts of 4A and 4B. If the part
4A is used as a memory for on-line use and the other part 4B is
used as the image file 10, the data transfer time required for
15 transfer to or recovery from the image file 10 can be much shorter.
Accordingly, the calculator initialization time of the whole
system can also be much shorter.

Further, as shown in FIG. 4, both of the image file 10 in
the main storage 4 and another image file 10 in an auxiliary
20 storage 3 may be used, and the image file 10 in the main storage 4
may be used usually. Then, even when the image file 10 in the
main storage 4 is broken due to a hardware malfunction or any
other problem, the calculator can be initialized at a high speed
by using the latter image file 10 in the auxiliary storage 3.
25 This leads to improvements in performance as well as in
reliability.



⑯ 日本国特許庁(JP)

⑰ 特許出願公開

⑱ 公開特許公報(A)

平3-265027

⑲ Int. Cl.⁵

G 06 F 9/06
1/24

識別記号 庁内整理番号
410 B 7927-5B

⑳ 公開 平成3年(1991)11月26日

7832-5B G 06 F 1/00 350 A
審査請求 未請求 請求項の数 1 (全5頁)

㉑ 発明の名称 計算機システムの初期化装置

㉒ 特 願 平2-62650

㉓ 出 願 平2(1990)3月15日

㉔ 発明者 山崎 雅裕 東京都府中市東芝町1番地 株式会社東芝府中工場内

㉕ 出願人 株式会社東芝 神奈川県川崎市幸区堀川町72番地

㉖ 代理人 弁理士 紋田 誠

明細書

1. 発明の名称

計算機システムの初期化装置

2. 特許請求の範囲

初期プログラムローダを用いて補助記憶装置のロードモジュールファイルに保存されている各種プログラムと、データファイルに保存されている各種データを主記憶装置に順次格納していくことにより計算機システムの初期化処理を実行する計算機システムの初期化装置において、前記初期化完了時の前記主記憶装置の内容をそっくり保存するイメージメモリと、このイメージメモリに保存されている初期化内容を変更する必要の有無を判断する判断手段と、この判断手段により初期化内容変更必要と判断された場合に前記初期化処理を実行し完了時の前記主記憶装置の内容をそっくり前記イメージファイルに更新保存するイメージファイル保存手段と、前記判断手段により初期化内容変更必要無しと判断された場合に前記イメージファイルに保存されている内容を前記主記

憶装置に格納する手段とを備えていることを特徴とする計算機システムの初期化装置。

3. 発明の詳細な説明

[発明の目的]

(産業上の利用分野)

本発明は、計算機システムにおける初期化装置に関する。

(従来の技術)

一般に計算機システムを初期化する場合には、基本ソフトウェア(以下OSと称す)の初期化と、応用ソフトウェア(以下APLと称す)との初期化が実施される。OSの初期化では、OS基本部の主記憶装置上への常駐化、システムタスクの生成及びシステムタスクを実行する為の環境の設定等が実施される。また、APLの初期化でもAPL基本部の主記憶装置への常駐化、応用タスクの生成及び応用タスクを実行する為の環境設定等が実施される。その後、システムタスク及び応用タスクが起動されプログラミングされた業務を実施する。

ここで、タスクの生成とはロードモジュールサ

イズ、走行エリア、優先度等のいわゆるタスク情報を主記憶装置上にビルドアップすることで、通常はオブジェクトモジュールをリンクエージ編集した時にロードモジュールと共に出来るロードモジュールヘッダを基に生成することを言う。また、実行環境の設定とは、システムタスク及び応用タスクがアクセスするファイルの生成、制御テーブルの生成を言い、これも単に補助記憶装置から転送するだけでなく、初期化処理で主記憶装置上にビルドアップする処理を指す。

(発明が解決しようとする課題)

このように従来の計算機システムにおいては、ソフトウェアの主記憶装置への常駐化、タスクの生成、実行環境の設定等のため、初期化には多くの時間がかかり、種々の問題が生じていた。特にプロセス計算機システムでは、ハードウェア等の一過的な障害により再起動をかける場合がある。この時、システムの初期化時間が長いと、プラントプロセス量をリアルタイムに監視することが出来ず、計算機停止期間中、つまり、システム初期

(作用)

通常オペレータが計算機を初期化する場合は、クイックモードとなり、また計算機システムに障害が発生して計算機システムが再起動された場合にもクイックモードが選択される。しかし、計算機システムの構成要素が変更になった場合のみ、セーピングモードが選択される。

セーピングモードでは従来の初期化処理を実施し、その後初期化した主記憶状態をイメージファイルに転写する。クイックモードではセーピングモードで転写したイメージファイルの内容を主記憶装置上に逆転写する。

こうすることによりクイックモードでは従来の初期化処理をバイパス出来る為、通常の計算機初期化時においては、従来の計算機初期化時に比べて大幅に時間短縮が可能となる。

(実施例)

以下、本発明の実施例を添付図面を参照しながら詳細に説明する。

第1図は本発明の一実施例による計算機シス

テムのデータが欠落する。このため、オペレータによるデータ補正又は計算機内部での自動補正等を必要とし、プラントプロセス監視のリアルタイム性が失われる問題点があった。

そこで本発明は上記の問題点を解決して、短時間で計算機システムの初期化を完了し得る初期化装置を提供することを目的とする。

[発明の構成]

(課題を解決するための手段)

本発明は、計算機の初期化手段にシステムの構成要素を意識させながら初期化し、初期化完了した主記憶装置のイメージをイメージファイル(主記憶装置全領域の内容が物理的にコピーされた領域)に保存するセーピングモードと、前記セーピングモードで保存したイメージファイルを再度主記憶装置に復元してシステムを走行させるクイックモードとを具備し、通常の計算機初期化ではクイックモードを選択し、システム構成要素が変更になった場合のみ、セーピングモードが選択される様にしたことを特徴とするものである。

ム初期化装置のブロック構成図を示したものである。

同図において、中央処理装置100内に設けられる複数の演算制御装置1は共通バス2を介してIPL(初期プログラムロード)装置3、主記憶装置4、および、補助記憶装置5に接続されている。

主記憶装置4には、補助記憶装置5に格納されているロードモジュールファイル6を基に生成されたプログラム7と、データファイル8を基に生成されたデータ9とが格納されると共に、これらは後に演算制御装置1との間で基本ソフトウェア(オペレーティングシステム)を介してやりとりされる。

イメージファイル10は、基本ソフトウェア(オペレーションシステム)と応用ソフトウェア(アプリケーションシステム)との全プログラムの初期化完了時における主記憶装置4の全内容をコピーイメージで保存するデータファイルである。

IPL装置3には、オペレータからの初期化要求又は計算機障害発生時の自動再起動を実行するマイクロプログラムが格納されている。

この構成で、計算機システムの初期化処理は第2図のフローチャートに示す如く行われる。

即ち、まずオペレータ1からの初期化要求又は計算機障害発生時IPL装置3が動作する。

IPL装置3ではオペレータがシステム構成要素を変更した後の初期化要求か否かを制御(処理101)し、システム構成要素変更後の初期化要求時には、イメージファイル10を無効とする(処理102)。次に、イメージファイル10が有効か否かを判定し(処理103)し、最初はイメージファイル10が処理102により無効とされているため、補助記憶装置5内のロードモジュールファイル6に記憶されている各プログラムおよびデータファイル8に記憶されている各種データを順次主記憶装置4のプログラム格納エリア7およびデータ格納エリア9に記憶するOSの初期化処理(処理104)およびアプリケーションの初期化処理(処理105)を実施する。OSの初期化処理及びアプリケーションの初期化処理完了後、システムを一旦オンライン状態よりオフライン状態に移し、スタティックな状態にした後、

このように、オペレータのセービングモード要求での計算機初期化要求か否かを判断し、セービングモードでの初期化要求時には従来の初期化処理を実施する。その、初期化完了時点で主記憶装置4のイメージをイメージファイル10に保存しておき、次回からはイメージファイル10が有効か否かを判断する。この結果、イメージファイル有効の場合には、従来の初期化処理をバイパスし、保存しておいたイメージファイル10の内容を主記憶装置4にそっくり移して計算機を初期化する。これによって、計算機の初期化時間を従来のものと比べて大幅に短縮出来る。つまりは、第2図のOSの初期化処理104とアプリケーションの初期化処理105と言った本来の初期化処理をバイパスして單にイメージファイル10の内容を主記憶装置4に転送するだけで初期化が可能となるため、従来の初期化に比べて格段に高速になることは明白である。

尚、以上の実施例ではイメージファイル10は補助記憶装置3に設けた例について説明したが、第3

主記憶装置4の全内容をイメージファイル10に出力する(処理106)。その後、イメージファイル10を有効とし(処理107)、IPL装置3に対して再起動要求を出す(処理108)。

これにより、IPL装置3ではセービングモードか否か再度判定し(処理101)、今回はセービングモードではない為、イメージファイル10を無効とする処理(処理102)はバイパスする。次にイメージファイル10が有効か否か判定するが(処理103)、以前の処理107でイメージファイル10は有効となっている為、イメージファイル10の内容を主記憶装置4に出力する(処理109)。その後、システムはオンラインとなり、バッチ処理及びリアルタイム処理が可能となる。

つまり、第2図においてOSの初期化処理104の前(ポイント201)で一旦オンラインとなり、アプリケーションの初期化処理105の後(ポイント202)で再度オフラインとなる。又イメージファイル10の内容を主記憶装置4に復元(処理109)した後(ポイント203)でオンラインとなる。

図に示すように主記憶装置4を4A,4Bに2分割し、4Aをオンライン用で使用するメモリ、4Bをイメージファイル10として使用すればイメージファイル10への転送又はイメージファイル10からの復元に要するデータ転送時間が更に短くなり、システム全体としての計算機初期化時間も更に短縮出来る。

更には第4図に示すように、主記憶装置4でのイメージファイル10と、補助記憶装置3でのイメージファイル10と、両方使用することにより通常は主記憶装置4のイメージファイル10を使用し、主記憶装置4がハードウェア的又はそれ以外の何らかの障害でイメージファイル10が破壊された場合でも、補助記憶装置3のイメージファイル10を使用することにより計算機を高速に初期化出来る。ひいては信頼性を向上しながら性能向上につながる。

[発明の効果]

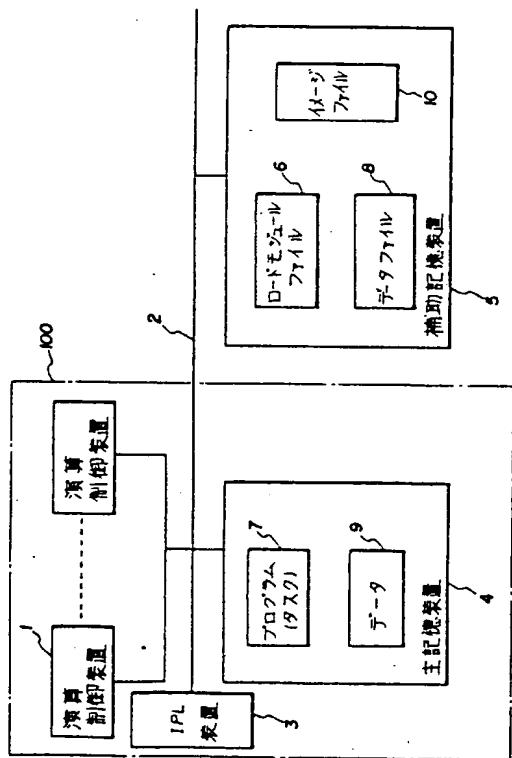
以上説明したように本発明によれば、オペレータは一回のみ時間のかかるセービングモードで

システムを初期化しておけば、以降システム成要素が変わらない限り、以前の初期化イメージで計算機を初期化する為、従来の初期化時間と比べて大幅に時間短縮出来る。又計算機障害時システムが停止した後も初期化時間が速い為、リアルタイムデータの欠落時間は短かくデータ修正及び補正の手間が軽くなる。場合によっては無視出来る。又、プロセス計算機の場合には時々刻々変化するプラントプロセス量の監視を計算機ハードウェアの一過的な不具合により計算機が停止しても再起動時間が短かい為、監視不能つまりめくら運転の時間を短縮出来る等の優れた効果が得られる。

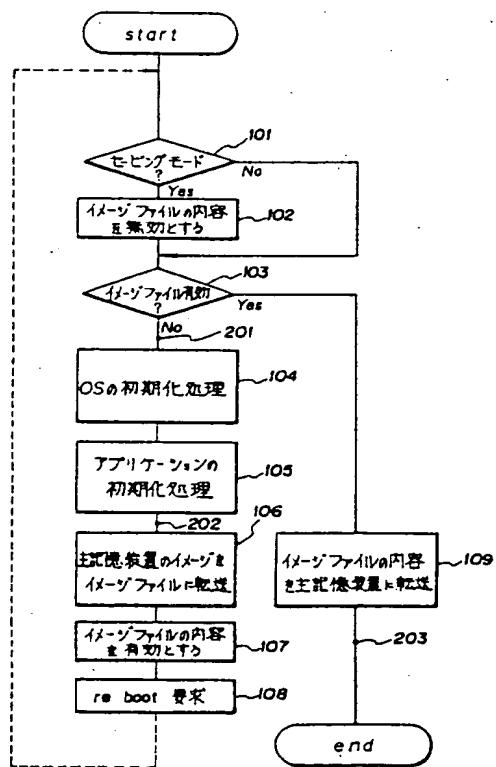
4. 図面の簡単な説明

第1図は本発明の一実施例における計算機システムブロック構成図、第2図は第1図の計算機システムの初期化処理の流れ図、第3図および第4図は本発明の他の実施例による計算機システムのブロック構成図である。

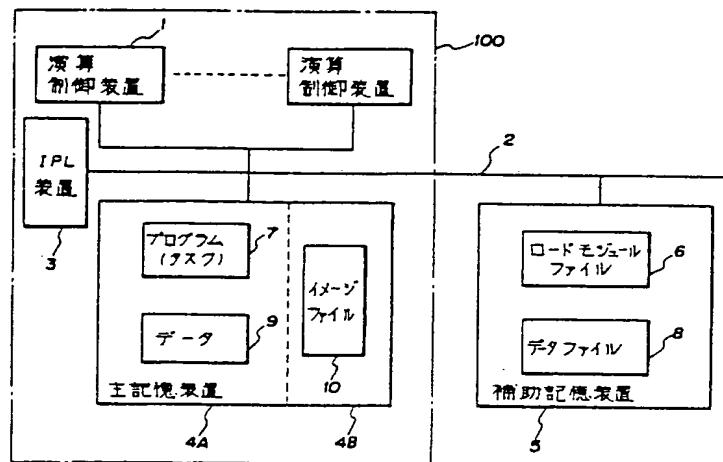
4…主記憶装置、5…補助記憶装置、6…ロードモジュールファイル、8…データファイル、



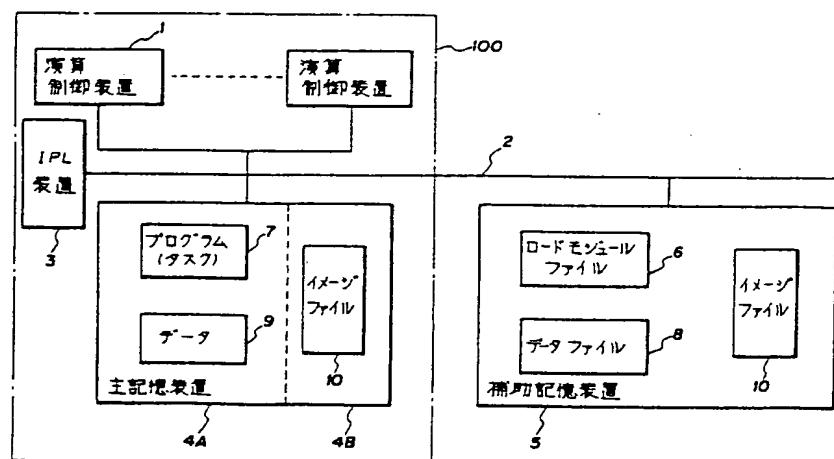
第一図



第二図



第3図



第4図